Week 2: Insights of Robustness

WEIRAN HUANG

Beijing University of Posts and Telecommunications

30 juillet 2019

Week 2 Outlines

Adversarial Examples on Graph Data

Deep Insight into Attack and Defense

Insights :

- **Perturbing edges** is more effective than modifying the features.
- The attack approaches tend to favor **adding edges** over removing.
- **Nodes with more neighbors are more difficult to atack.**

Adversarial Examples on Graph Data

Deep Insight into Attack and Defense

Insights :

- **Perturbing edges** is more effective than modifying the features.
- The attack approaches tend to favor **adding edges** over removing.
- **Nodes with more neighbors are more difficult to atack.**

Adversarial Examples on Graph Data
Deep Insight into Attack and Defense

Insights :

- **Perturbing edges** is more effective than modifying the features.
- The attack approaches tend to favor **adding edges** over removing.
- **Nodes with more neighbors are more difficult to atack.**

Adversarial Examples on Graph Data

Defense Techniques

Mothods :

- **make the adjacency matrix trainable : learn edge weights**. The model will assign lower weight to edges that connect dissimilar nodes.
- **Pre-processing** : Remove edges that connects nodes with low similarity score(=0 in their practice). It is more efficient because no extra parameters are introduced.

Adversarial Examples on Graph Data

Defense Techniques

Mothods :

- **make the adjacency matrix trainable : learn edge weights**. The model will assign lower weight to edges that connect dissimilar nodes.
- **Pre-processing** : Remove edges that connects nodes with low similarity score(=0 in their practice). It is more efficient because no extra parameters are introduced.

Why GCNs are vulnerable ?

Thoughts based on the paper

Limitations of Graph Convolution :

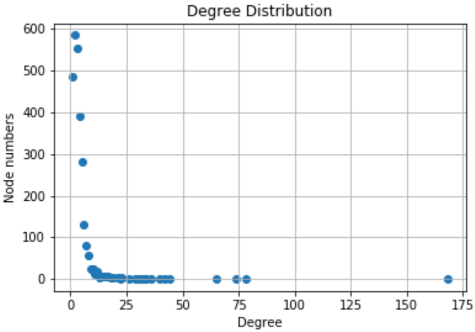
- **Power Law Distribution**. Most nodes have very few neighbors.

Why GCNs are vulnerable ?

Thoughts based on the paper

Limitations of Graph Convolution :

- **Power Law Distribution**. Most nodes have very few neighbors.

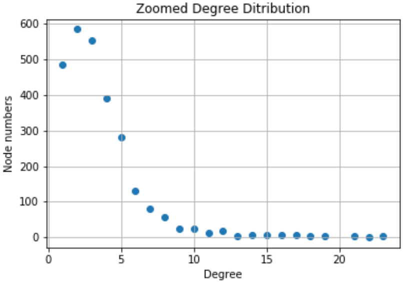


Why GCNs are vulnerable ?

Thoughts based on the paper

Limitations of Graph Convolution :

- **Power Law Distribution**. Most nodes have very few neighbors.



Nodes with degree no larger than 5 contributes 84.6% to the whole dataset Cora(2291/2708).

Why GCNs are vulnerable ?

Thoughts based on the paper

Limitations of Graph Convolution :

- **Power Law Distribution**. Most nodes have very few neighbors.
  Comparing with Insight 3 : Most nodes are suffering from **lack of information**, so they are vulnerable to noise.
- **Limitation of Local Aggregation**. Does a node necessarily need to be similar to its immediate neighbors ? Recall the classic network embedding model : **LINE**.

Why GCNs are vulnerable ?

Thoughts based on the paper

Limitations of Graph Convolution :

- **Power Law Distribution**. Most nodes have very few neighbors.
  Comparing with Insight 3 : Most nodes are suffering from **lack of information**, so they are vulnerable to noise.
- **Limitation of Local Aggregation**. Does a node necessarily need to be similar to its immediate neighbors ?
  Recall the classic network embedding model : **LINE**.

LINE

Large-scale Information Network Embedding

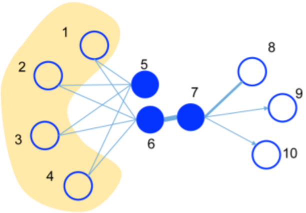


Figure 1: A toy example of information network. Edges can be undirected, directed, and/or weighted. Vertex 6 and 7 should be placed closely in the low-dimensional space as they are connected through a strong tie. Vertex 5 and 6 should also be placed closely as they share similar neighbors.

LINE considers both first-order and second-order similarity.

LINE

Large-scale Information Network Embedding

According to

$$Z = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta$$

we know by stacking 2 GCNs layers, nodes can aggregate information from second-order neighbors, but through normalization, the impact is quite small and involves a lot of noises(from first-order neighbors), rendering it more likely to over-smooth.

LINE

Large-scale Information Network Embedding

```
label:     0
neighbor:    [410, 963, 2120, 2753, 3333, 4237, 9389, 9409, 9910, 10256, 11084, 13261, 13549, 17701, 18843, 18988]
neighbor labels:    [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2]

label:     2
neighbor:    [282, 7154, 8661, 8886, 9743, 11369, 11668, 11915, 13515, 14022, 14769, 15632, 16369, 18356, 18967]
neighbor labels:    [2, 2, 0, 2, 2, 2, 2, 2, 2, 0, 0, 2, 2, 2, 0]

label:     1
neighbor:    [15479]
neighbor labels:    [1]

label:     1
neighbor:    [1321, 11076]
neighbor labels:    [0, 1]

label:     0
neighbor:    [6077, 6689]
neighbor labels:    [2, 2]
```

Our Model

To achieve robustness

Goals :

- Ensure the number of nodes in a single aggregation.
- Explicitly involve higher-order(especially second-order) neighbors to a GCN layer(not by stacking).

Our Model

To achieve robustness

Goals :

- Ensure the number of nodes in a single aggregation.
- Explicitly involve higher-order(especially second-order) neighbors to a GCN layer(not by stacking).

Our Model
To achieve robustness

A naive approach :

1. Replace the adjcency matrix A with $A + A^2$. (Introduce more edges and higher-order neighbors)
2. Then follow the method in the previous paper, remove the edges which connect dissimilar nodes(Jaccard index=0).(remove possible noises)

Our Model

To achieve robustness

More complex methods(NOT YET INVESTIGATED) :

1. Dilated convolution in GNN.(see next page) from *Can GCNs Go as Deep as CNNs ?*.
2. KDD 18 : *GeniePath : Graph Neural Networks with Adaptive Receptive Paths*.

Dilated Convolution

Can GCNs Go as Deep as CNNs ?

Dynamic Edges. As mentioned earlier, most GCNs only update the vertex features at each iteration. Recent works [35, 43, 39] show that dynamic graph convolution can learn better graph representations compared to GCNs with fixed graph structures. For instance, ECC (Edge-Conditioned Convolution) [35] uses dynamic edge-conditional filters to learn an edge-specific weight matrix. EdgeConv [43] finds the nearest neighbors in the feature space to reconstruct the graph after every EdgeConv layer. In order to learn to generate point clouds, Graph-Convolution GAN (Generative Adversarial Network) [39] also applies k-NN graphs to construct the neighbourhood for each vertex in every layer. We find that dynamically changing neighbors of GCNs helps to alleviate the over-smoothing problem and results in an effectively larger receptive field. In our framework, we propose to re-compute edges between vertices via a *Dilated k-NN* in the feature space at each layer to further increase the receptive field.

Dilated Convolution

Can GCNs Go as Deep as CNNs ?

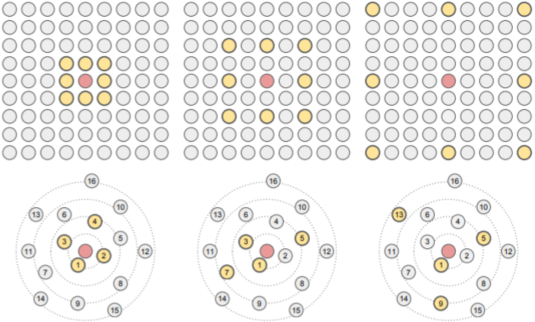


Figure 3. **Dilated Convolutions in GCNs.** Visualization of dilated convolution on a structured graph arranged in a grid (*e.g.* 2D image) and on a general structured graph. *Top:* 2D convolution with kernel size 3 and dilation rate 1, 2, 4 (left to right). *Bottom:* Dynamic graph convolution with dilation rate 1, 2, 4 (left to right).

GeniePath

Adaptive Receptive Path

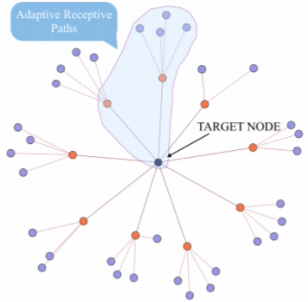


Figure 2: A motivated illustration of meaningful receptive paths (shaded) given all two-hops neighbors (red and blue nodes), with the black node as target node.

GeniePath

Adaptive Receptive Path

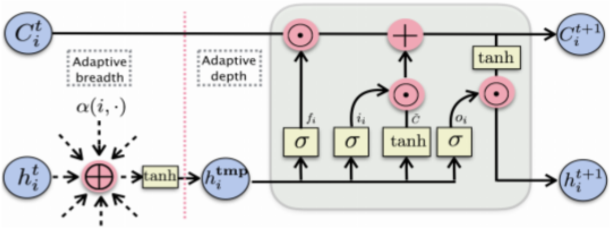


Figure 3: A demonstration for the architecture of GeniePath. Symbol $\bigoplus$ denotes the operator $\sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha(h_i^{(t)}, h_j^{(t)}) \cdot h_j^{(t)}$.

Further Discussion

From Defense Perspective

Why would this model possibly work ?

- We focus on defense against such attacks that only add/remove edges. Our model aims to introduce more informative edges and downweight useless edges, so theoretically it should defense attack and even outperform current GCN models with clean data.

  Remark. In Dai's *Adversarial Attack on Graph Structured Data*, they limit both the number of added/removed edges and the **original distance** between the newly connected nodes. If such edges have already been considered in our model, the perturbation could be minimized.

Further Discussion

From Defense Perspective

Remaining work :

- How to aggregate higher-order neighbor information ?
- How to downweight or prune the edges ?